AMENDMENTS

In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("—") being deleted:

1. (Currently Amended) A method of monitoring response time of a ~~method or a~~ function associated with a software component, comprising:

operating on a bytecode representation of a ~~method or~~ function to be instrumented by inserting an instrumentation code in the bytecode representation of ~~said method or~~ the function without modifying ~~the~~ respective source code of ~~said method or~~ the function and while classes of ~~said method or~~ the function are being loaded for ~~execution;~~ execution and incorporating instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;

generating a call to an Application Response Measurement (ARM) agent to cause the ARM agent to effect generation of a start time marker upon start of execution of ~~said method or~~ the function and a stop time marker upon completion of execution of ~~said method or~~ the function, wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

utilizing ~~said~~ the start and stop time markers to determine a response time of ~~said method or~~ the function.

2. (Canceled)

3. (Currently Amended) The method of claim 1, further comprising:

registering ~~said method or~~ the function with ~~said~~ the ARM agent prior to generation of ~~said~~ the start and stop time markers.

4. (Currently Amended) The method of claim 1, wherein ~~said~~ the instrumentation code causes generation of ~~said~~ the start and stop time markers without modifying instructions associated with execution of ~~said method or~~ the function.

5. (Currently Amended) The method of claim 1, wherein ~~said~~ the ARM agent generates a record corresponding to ~~said method or~~ the function for storing the response time associated with ~~said method or~~ the function.

6. (Currently Amended) The method of claim 5, wherein ~~said~~ the record includes a field for identifying a parent, if any, of ~~said method or~~ the function in a hierarchical parent-child transaction chain.

7. (Currently Amended) The method of claim 6, wherein ~~said~~ the record includes another field for identifying a top level transaction in ~~said~~ the parent-child transaction chain.

8. (Currently Amended) The method of claim 1, wherein ~~said~~ the software component ~~can be any of~~ includes at least one of the following: a server page, a serylet of a server side component, a driver, a naming and directory interface (NDI) or remote method invocation (RMI) component.

9. (Currently Amended) The method of claim 8, wherein ~~said method or~~ the function of the software component ~~comprises any of a Service~~ includes at least one of the following: a service method of a server page, a doFilter, a doGet, a doPost or a service method of a serylet, a getConnection, executeQuery, or selected methods of driver, or remote, local or home interface methods of a server side component.

10. (Canceled)

11. (Currently Amended) The method of claim 1, further comprising:

storing ~~said~~ the response time in a database.

12. (Currently Amended) The method of claim 1, further comprising:

displaying ~~said~~ the response time to a user.

13. (Currently Amended) A system ~~comprising~~ comprising:

a memory component; and

a processor configured to monitor a response time of a ~~method or~~ function associated with a software component, ~~said~~ the processor configured to ~~implement at least:~~ implement:

an instrumentation engine for operating on a bytecode representation of a ~~method or~~ function to be instrumented by inserting instrumentation code in the bytecode representation of ~~said method or~~ the function without modifying ~~the~~ respective source code of ~~said method or~~ the function and while classes of ~~said method or~~ the function are being loaded for execution, ~~said~~ the instrumentation code effecting generation of a start time marker and a stop time marker upon resumption and completion, respectively, of ~~said method or function,~~ the function, the instrumentation code further configured to incorporate instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;

an interface module being invoked by ~~said~~ the instrumentation code upon start and completion of ~~said method or function,~~ the function;

an application response measurement (ARM) agent in communication with ~~said~~ the interface ~~module,~~ module;

wherein ~~said~~ the interface module, upon invocation by ~~said~~ the instrumentation code, calls ~~said~~ the ARM agent to cause generation of ~~said~~ the start and stop time markers by ~~said~~ the ARM agent, and wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

an analysis and presentation module in communication with ~~said~~ the ARM agent for presenting ~~said~~ the response time to a user and/or storing ~~said~~ the response time in a database.

14. (Currently Amended) The system of claim 13, wherein ~~said~~ the instrumentation engine inserts ~~said~~ the instrumentation code prior to loading of a class containing ~~said method or~~ the function by a virtual machine.

15. (Currently Amended) The system of claim 13, wherein ~~said~~ the instrumentation engine inserts ~~said~~ the instrumentation code in ~~said~~ the bytecode representation without modifying instructions associated with execution of ~~said method or~~ the function.

16. – 21. (Canceled).

22. (New) A system of monitoring response time of a function associated with a software component, comprising:

means for operating on a bytecode representation of a function to be instrumented by inserting an instrumentation code in the bytecode representation of the function without modifying respective source code of the function and while classes of the function are being loaded for execution and incorporating instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;

means for generating a call to an Application Response Measurement (ARM) agent to cause the ARM agent to effect generation of a start time marker upon start of execution of the function and a stop time marker upon completion of execution of the function, wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

means for utilizing the start and stop time markers to determine a response time of the function.

23. (New) The method of claim 22, further comprising:

means for registering the function with the ARM agent prior to generation of the start and stop time markers.

24. (New) The method of claim 22, further comprising means for causing generation of the start and stop time markers without modifying instructions associated with execution of the function.

25. (New) The method of claim 22, further comprising means for a record corresponding to the function for storing the response time associated with the function.

26. (New) The method of claim 25, wherein the record includes a field for identifying a parent, if any, of the function in a hierarchical parent-child transaction chain.

27. (New) The method of claim 26, wherein the record includes another field for identifying a top level transaction in the parent-child transaction chain.

28. (New) The method of claim 22, wherein the software component includes at least one of the following: a server page, a serylet of a server side component, a driver, a naming and

directory interface (NDI) or remote method invocation (RMI) component.

29. (New) The method of claim 28, wherein the function of the software component includes at least one of the following: a service method of a server page, a doFilter, a doGet, a doPost or a service method of a serylet, a getConnection, executeQuery, or selected methods of driver, or remote, local or home interface methods of a server side component.